CHAPTER 1
INTRODUCTION

1.1 Overview of CVD

Cardiovascular disease (CVD) is increasing day by day in this modern world. According to WHO, an estimated 17 million people die of CVDs particularly heart attack and strokes, every year. Thus, it is necessary to enlist the most important symptoms and health habits which contribute towards CVDs.

Different tests are conducted before diagnosing CVD, Some of them are auscultation, ECG, blood pressure, cholesterol and blood sugar. These tests are often extensive and time consuming while a patient's health condition may be critical and he/she needs to start immediate medication so it becomes important to prioritize the tests. There are several health habits which contribute to CVDs. Therefore, it is also necessary to know which of the health habits contribute to CVDs so that healthy health habits can be practiced.

While cardiovascular disease can refer to different heart or blood vessel problems, the term is often used to mean damage to your heart or blood vessels by atherosclerosis, a build-up of fatty plaques in your arteries. Plaque buildup thickens and stiffens artery walls, which can inhibit blood flow through your arteries to your organs and tissues.

1.2 Symptoms

Heart disease symptoms depend on what type of heart disease you have.

Symptoms of heart disease in your blood vessels (atherosclerotic disease)

Cardiovascular disease symptoms may be different for men and women. For instance, men are more likely to have chest pain; women are more likely to have other symptoms along with chest discomfort, such as shortness of breath, nausea and extreme fatigue.

Symptoms can include:

- Chest pain, chest tightness, chest pressure and chest discomfort (angina)
- Shortness of breath
- Pain, numbness, weakness or coldness in your legs or arms if the blood vessels in those parts of your body are narrowed
- Pain in the neck, jaw, throat, upper abdomen or back

You might not be diagnosed with cardiovascular disease until you have a heart attack, angina, stroke or heart failure. It's important to watch for cardiovascular symptoms and discuss concerns with your doctor. Cardiovascular disease can sometimes be found early with regular evaluations.

Heart disease symptoms caused by abnormal heartbeats (heart arrhythmias)

A heart arrhythmia is an abnormal heartbeat. Your heart may beat too quickly, too slowly or irregularly. Heart arrhythmia symptoms can include:

- Fluttering in your chest
- Racing heartbeat (tachycardia)

- Slow heartbeat (bradycardia)
- Chest pain or discomfort
- Shortness of breath
- Lightheadedness
- Dizziness
- Fainting (syncope) or near fainting

Heart disease symptoms caused by heart defects

Serious congenital heart defects — defects you're born with — usually become evident soon after birth. Heart defect symptoms in children could include:

- Pale gray or blue skin color (cyanosis)
- Swelling in the legs, abdomen or areas around the eyes
- In an infant, shortness of breath during feedings, leading to poor weight gain

Less serious congenital heart defects are often not diagnosed until later in childhood or during adulthood. Signs and symptoms of congenital heart defects that usually aren't immediately life-threatening include:

- Easily getting short of breath during exercise or activity
- Easily tiring during exercise or activity
- Swelling in the hands, ankles or feet

Heart disease symptoms caused by weak heart muscle (dilated cardiomyopathy)

In early stages of cardiomyopathy, you may have no symptoms. As the condition worsens, symptoms may include:

- Breathlessness with exertion or at rest
- Swelling of the legs, ankles and feet
- Fatigue
- Irregular heartbeats that feel rapid, pounding or fluttering
- Dizziness, lightheadedness and fainting

A beating heart contracts and relaxes in a continuous cycle.

- During contraction (systole), your ventricles contract, forcing blood into the vessels to your lungs and body.
- During relaxation (diastole), the ventricles are filled with blood coming from the upper chambers (left and right atria).

Causes of cardiomyopathy include

The cause of cardiomyopathy, a thickening or enlarging of the heart muscle, may depend on the type:

- **Dilated cardiomyopathy** The cause of this most common type of cardiomyopathy often is unknown. It may be caused by reduced blood flow to the heart (ischemic heart disease) resulting from damage after a heart attack, infections, toxins and certain drugs. It may also be inherited from a parent. It usually enlarges (dilates) the left ventricle.

- **Hypertrophic cardiomyopathy** This type, in which the heart muscle becomes abnormally thick, usually is inherited. It can also develop over time because of high blood pressure or aging.
- **Restrictive cardiomyopathy** This least common type of cardiomyopathy, which causes the heart muscle to become rigid and less elastic, can occur for no known reason. Or it may be caused by diseases, such as connective tissue disorders, excessive iron buildup in your body (hemochromatosis), the buildup of abnormal proteins (amyloidosis) or by some cancer treatments.

Causes of heart infection include

A heart infection, such as endocarditis, is caused when an irritant, such as a bacterium, virus or chemical, reaches your heart muscle. The most common causes of heart infection include:

- Bacteria
- Viruses
- Parasites

Risk factors for developing heart disease include:

- **Age** Aging increases your risk of damaged and narrowed arteries and weakened or thickened heart muscle.
- **Sex** Men are generally at greater risk of heart disease. However, women's risk increases after menopause.

- **Family history** A family history of heart disease increases your risk of coronary artery disease, especially if a parent developed it at an early age (before age 55 for a male relative, such as your brother or father, and 65 for a female relative, such as your mother or sister).

- **Smoking** Nicotine constricts your blood vessels, and carbon monoxide can damage their inner lining, making them more susceptible to atherosclerosis. Heart attacks are more common in smokers than in nonsmokers.

- **Certain chemotherapy drugs and radiation therapy for cancer** Some chemotherapy drugs and radiation therapies may increase the risk of cardiovascular disease.

- **Poor diet** A diet that's high in fat, salt, sugar and cholesterol can contribute to the development of heart disease.

- **High blood pressure** Uncontrolled high blood pressure can result in hardening and thickening of your arteries, narrowing the vessels through which blood flows.

- **High blood cholesterol levels** High levels of cholesterol in your blood can increase the risk of formation of plaques and atherosclerosis.

- **Diabetes** Diabetes increases your risk of heart disease. Both conditions share similar risk factors, such as obesity and high blood pressure.

- **Obesity** Excess weight typically worsens other risk factors.

- **Physical inactivity** Lack of exercise also is associated with many forms of heart disease and some of its other risk factors, as well.

- **Stress** Unrelieved stress may damage your arteries and worsen other risk factors for heart disease.

- **Poor hygiene** Not regularly washing your hands and not establishing other habits that can help prevent viral or bacterial infections can put you at risk of heart infections, especially if you already have an underlying heart condition. Poor dental health also may contribute to heart disease.

Complications of heart disease include

- **Heart failure** One of the most common complications of heart disease, heart failure occurs when your heart can't pump enough blood to meet your body's needs. Heart failure can result from many forms of heart disease, including heart defects, cardiovascular disease, valvular heart disease, heart infections or cardiomyopathy.

- **Heart attack** A blood clot blocking the blood flow through a blood vessel that feeds the heart causes a heart attack, possibly damaging or destroying a part of the heart muscle. Atherosclerosis can cause a heart attack.

- **Stroke** The risk factors that lead to cardiovascular disease also can lead to an ischemic stroke, which happens when the arteries to your brain are narrowed or blocked so that too little blood reaches your brain. A stroke is a medical emergency — brain tissue begins to die within just a few minutes of a stroke.

- **Aneurysm** A serious complication that can occur anywhere in your body, an aneurysm is a bulge in the wall of your artery. If an aneurysm bursts, you may face life-threatening internal bleeding.

- **Peripheral artery disease** Atherosclerosis also can lead to peripheral artery disease. When you develop peripheral artery disease, your extremities — usually your legs — don't receive enough blood flow. This causes symptoms, most notably leg pain when walking (claudication).
- **Sudden cardiac arrest** Sudden cardiac arrest is the sudden, unexpected loss of heart function, breathing and consciousness, often caused by an arrhythmia. Sudden cardiac arrest is a medical emergency. If not treated immediately, it is fatal, resulting in sudden cardiac death.

1.3 Objective

The Objective of our project is to improve the heart disease classification using machine learning algorithm to enhance the survival rate of the human.

1.4 Overview of Machine Learning

Machine learning is an emerging field today due to a rise in the amount of data. Machine learning helps to gain insight from a massive amount of data which is very cumbersome to humans and sometimes also impossible. The study's objective is to prioritize the diagnosis test and see some of the health habits which contribute to CVD. Furthermore, and most importantly, different machine learning algorithms are compared according to different performance metrics. In this thesis, manually classified data is used. Manual classification is either healthy or unhealthy. Based on a machine learning technique called classification, 70 percent data are supervised or trained and 30 percent are tested in this thesis. Thus, different algorithms are compared as per their prediction results. Machine learning is a subfield of computer science and a rapidly up surging topic in today's context and is expected to boom more in coming days. Our world is flooded with data and data is being created rapidly every day all around the world. According to Big Data and Analytics Solutions company CSC, it is expected by 2020, that the data amount will be 44 times bigger than in 200 9. Therefore, it is necessary to understand data and gain insights for better understanding of a human world. The data amount is so huge today that traditional methods cannot be used. Analysing data or building predictive models manually is almost impossible in some scenarios and also time consuming and less productive. Machine learning, on other hand, produces reliable, repeatable results and learns from earlier computation.

Data used for machine learning are basically of two types labelled data and unlabelled data. Labelled data is the data where attributes are provided. It has some sort of tag or meaning attached to the data therefore used in supervised learning. Labelled attribute can be numerical or categorical. Numerical data are used in regression to predict the value while categorical data are used in classification. Unlabelled data is the data where there are only data points and no labelling to assist. Unlabelled data are used in unsupervised learning so that machine can identify the patterns or any structure present in the data set.

The labelled data and unlabelled data are used with supervised learning and unsupervised learning respectively. Supervised learning entails a learning map between a set of input variables X and an output variable Y and applying this mapping to predict the output for unseen data. After learning the dataset, algorithms generalise the data and formulates the hypothetical value H for the given dataset.

Supervised learning is further categorised into two types: Regression and Classification. According to the business dictionary, a regression is a technique for determining the statistical relationship between two or more variables where a change in dependent variable is associated with, and depends on a change in one or more independent variables. Classification is a task that occurs very frequently in everyday life. Essential- 3 ly it involves dividing up objects so that each is assigned to one of a number of mutually exhaustive and exclusive categories known as classes. The term ‘mutually exhaustive and exclusive’ simply means that each object must be assigned to precisely one class, i.e. never to more than one and never to no class at all.

Unsupervised learning studies how systems can learn to represent particular input patterns in a way that reflects the statistical structure of the overall collection of input patterns. By contrast, with supervised learning there are no explicit target outputs or environmental evaluations associated with each input; rather the unsupervised learner brings prior biases as to what aspects of the structure of the input should be captured in the output.

CHAPTER 2

Literature Survey

2.1 Related Works

There are a lot of studies on prediction of heart disease as a medical diagnosis system.

Firstly, R.W.Jones, et al. (2017) proposed a study applying neural network to self-applied questionnaire (SAQ) data to develop a heart disease prediction system. The study not only clarifies common risk factors of the disease but also the other data collected in SAQ. The validation of the work was provided by checking against the result of the neural network with "Dundee Rank Factor Score" which is related to statistically 3 risk factors (blood pressure, smoking and blood cholesterol) together with sex and age to determine risk of having heart disease. In the study, they used multi-layered feed forward neural network which was trained with Back propagation Algorithm. There were three layers in the neural network they used: input, hidden and output layers. The performance was improved to Relative Operating Characteristic (ROC) area of 98% by increasing input numbers of the neural network [16].

Aditi Gavhane et al. (2018) proposed neural network algorithm multi-layer perceptron (MLP) to develop an application which can predict the vulnerability of a heart disease given basic symptoms like age, sex, pulse rate etc. The machine learning algorithm neural networks has proven to be the most accurate and reliable algorithm and hence used in the proposed system [2].

Ankita Dewan et al. (2015) discussed various kinds of techniques for developing a heart disease prediction system and proposed using Backpropagation Algorithm as best classification technique for the targeted system. They also have proposed using Genetic Algorithm as optimizer against the Backpropagation Algorithm drawback of being stuck in local minima. The proposed methodology was intended for implementing in future with an accuracy of nearly 100% or with minimal error[4].

M.A.Jabbar et al. (2016) proposed model claims that the Hidden Naïve Bayes (HNB) can be applied to heart disease classification (prediction). Our experimental results on heart disease data set show that the HNB records 100% in terms of accuracy and out performs naïve Bayes [10].

Tülay KarayÕlan et al. (2017) proposed a heart disease prediction system which uses artificial neural network backpropagation algorithm is proposed.13 clinical features were used as input for the neural network and then the neural network was trained with backpropagation algorithm to predict absence or presence of heart disease with accuracy of 95% [20].

Rifki Wijaya et al. (2013) proposed that the development of heart disease prediction using machine learning (in this case the Artificial Neural Network or ANN). Prediction of a person's heart disease one year ahead is performed by studying the model heart rate data. Data is taken by using tool such as smart mirror, smart mouse, smart phones and smart chair. Heart rate data were collected through the Internet and collected in a server. Learning in this system is performed for a period of one year to get enough data to make predictions. Predictive of future heart disease in one year can increase a person's awareness of heart disease itself. The system is also expected to reduce the number of patients and the number of deaths from heart disease [15].

Aakash Chauhan et al.(2018) proposed a Weighted Association Rule is a type of data mining technique used to eliminate the manual task which also helps in extracting the data directly from the electronic records. This will help in decreasing the cost of services and also helps in saving lives. It finds the rule to predict patient's risk of having coronary disease. Test results have shown that vast majority of the rules helps in the best prediction of coronary illness [1].

A H Chen et al(2011) developed a heart disease predict system that can assist medical professionals in predicting heart disease status based on the clinical data of patients. It include three steps. Firstly, to select 13 important clinical features, i.e., age, sex, chest pain type, trestbps, cholesterol, fasting blood sugar, resting ecg, max heart rate, exercise induced angina, old peak, slope, number of vessels colored, and thal. Secondly, to develop an artificial neural network algorithm for classifying heart disease based on these clinical features. The accuracy of prediction is near 80%. Finally, developed a user-friendly heart disease predict system (HDPS). The HDPS system will be consisted of multiple features, including input clinical data section, ROC curve display section, and prediction performance display section (execute time, accuracy, sensitivity, specificity, and predict result) [3].

Sana Shaikh et al.(2015) proposed a system that intended to develop an Intelligent System using data mining modelling technique, namely, Naive Bayes. It is implemented as Java application in which user answers the predefined questions. It retrieves hidden data from stored database and compares the user values with trained data set. It can answer complex queries for diagnosing heart disease and thus assist healthcare practitioners to make intelligent clinical decisions which traditional decision support systems cannot [17].

Rashmi G Saboji et al. (2017) proposed a scalable framework that uses healthcare data to predict heart disease based on certain attributes and to predict the diagnosis of heart disease with a small number of attributes. The prediction solution uses random forest on Apache Spark, which gives massive opportunity for health care analysts to deploy this solution on ever changing, scalable big data landscape for insightful decision making. Using this approach, 98% accuracy is achieved. A comparison against Naïve-Bayes classifier, where we show the random forest approach outperforms the former by a significant margin [14].

Lalaantika Sharma et al. (2016) proposed a Classification and development of tool for heart disease (MRI images) using machine learning developed Computational method that has potential to predict disease in early stages automatically and especially helpful in resources limited countries. Computational method to predict global hyperkinesia based on confirms cases of global hyperkinesia through MRI was developed. Almost all feature extraction method was used on MRI images and model was generated on merged and different images separately. High accuracy of model independent test set justified our approaches and reliability of model. The newly developed was implemented in python and available for open use [9].

Tahira Mahboob et al. (2017) proposed a Evaluating ensemble prediction of coronary heart disease using receiver operating characteristics suggests that inferring heart disease has turn into foremost distress currently. It centers on various machine learning practices which assist ascertaining and perceiving innumerable heart diseases. Multifarious machine learning approaches conversed here are Hidden Markov Models, Support Vector Machine, Feature Selection, Computational intelligent classifier, prediction system, data mining techniques and genetic algorithm. Scrutinizing each approach thoroughly allowed us to select most apposite one. This ultimately permits us to propose an Ensemble Model exploiting pertinent machine learning procedures which perfectly categorizes diverse heart diseases. The evaluation of the proposed technique has been conducted using state of the art technology. The proposed technique has an accuracy of 94.21%, a ROC (Receiver Operating Characteristics) of 0.981, RMSE (Root Mean Square Error) of .2568, Precision of 0.953; showing significant improvement when compared to the performance of K-Nearest Neighbour, Artificial Neural Networks and Support Vector Machines algorithms. Analysis/Evaluation of the implemented algorithms and the proposed Ensemble Model has been done expending the Receiver Operator Characteristics [19].

N. Prabakaran et al. (2018) proposed a prediction of Cardiac disease Based on Patient's Symptoms developed spatial as well as temporal demographic, and symptom information is available at the data presented during the time of execution. Our proposed method incorporates all such information that is being used as a classification approach that compares recent healthcare data against data from that particular baseline distribution and hence classifies subgroups of the given data. In addition, the data sample data used is first tested against many types of classifiers and various other proposed test scores have been evaluated. Test best is further chosen to make predictions. This follows a prototype implementation using a python based data mining tool, Orange (version: 0.17.1). The database can be stored in a cloud to centralize it and make access easier [13].

Jayshril S. Sonawane et al. (2014) proposed a Prediction of heart disease using learning vector quantization algorithm suggests that in medical field the disease diagnosis is often made based on the knowledge and experience of the medical practitioner. Due to this there are chances of errors, unwanted biases and also takes longer time in accurate diagnosis of disease. In case of heart disease, its diagnosis is most difficult task. It depends on the careful analysis of different clinical and pathological data of the patient by medical experts, which is complicated process. Due to advancement in machine learning, computer and information technology, the researchers and medical practitioners in large extent are interested in the development of automated system for the prediction of heart disease. In this they present a prediction system for heart disease using Learning vector Quantization neural network algorithm. The neural network in this system accepts 13 clinical features as input and predicts that there is a presence or absence of heart disease in the patient, along with different performance measures [8].

Meenal Saini et al. (2017) proposed a Prediction of heart disease severity with hybrid data mining suggests that Nowadays, heart diseases are very common and one of the major causes of death across the globe. This calls for an accurate and timely diagnosis of the heart disease. There is abundant data available with the health care systems; however, the knowledge about the data is rather poor. The accessibility of the enormous size of medical dataset hints towards the requirement of a tool which analyses data to extract valuable information. Data scientists have attempted several methods in order to improvise the examination of large data sets. Previously, various data mining techniques have been implemented in the healthcare systems, however, the hybridization in addition to a single technique in the identification of heart disease shows promising outcomes, and can be useful in further investigating the treatment of the heart diseases. This work attempts to survey some recent techniques applied towards knowledge discovery for heart disease prediction and further proposes a novel prediction method with improved accuracy [11].

H. Mai et al. (2018) proposed a Non-Laboratory-Based Risk Factors for Automated Heart Disease Detection in 2018 suggests that developing a heart disease detection model using simple non-laboratory risk factors plays an important role in preventive care, especially for high risk subjects. The model allows physicians/epidemiologists to effectively diagnose a person as having heart disease. In this work, we aim to develop a non-invasive risk prediction model for automated heart disease detection that involves age, gender, rest blood pressure, maximum heart rate, and rest electrocardiography. They examine four public datasets from 1071 participants who were referred for a special X-ray of the heart's arteries (i.e., to see if they are narrowed or blocked). The subjects also undertook a physical examination and three non-invasive tests. To estimate the heart disease status, we apply a generalized linear model with regularization paths via coordinate descent. Even without laboratory-based data (e.g., serum cholesterol, fasting blood sugar), we observed a prediction accuracy as high as 72%, compared with 76% of other comprehensive models. This observation suggests that few non-invasive factors utilizing recent advances in data analytics can replace the current practices of heart disease risk assessment [6].

N. K. Salma Banu et al. (2016) proposed a Prediction of heart disease at early stage using data mining and big data analytics: A survey suggests that the various technologies of data mining (DM) models for forecast of heart disease are discussed. Data mining plays an important role in building an intelligent model for medical systems to detect heart disease (HD) using data sets of the patients, which involves risk factor associated with heart disease. Medical practitioners can help the patients by predicting the heart disease before occurring. The large data available from medical diagnosis is analysed by using data mining tools and useful information known as knowledge is extracted. Mining is a method of exploring massive sets of data to take out patterns which are hidden and previously unknown relationships and knowledge detection to help the better understanding of medical data to prevent heart disease. There are many DM techniques available namely Classification techniques involving Naïve Bayes (NB), Decision tree (DT), Neural network (NN), Genetic algorithm (GA), Artificial intelligence (AI) and Clustering algorithms like K-NN, and Support vector machine (SVM). Several studies have been carried out for developing prediction model using individual technique and also by combining two or more techniques. This paper provides a quick and easy review and understanding of available prediction models using data mining from 2004 to 2016. The comparison shows the accuracy level of each model given by different researchers [12].

S. M. M. Hasan et al. (2018) proposed a Comparative Analysis of Classification Approaches for Heart Disease Prediction suggests that Heart disease is one of the most common causes of death around the world nowadays. Often, the enormous amount of information is gathered to detect diseases in medical science. All of the information is not useful but vital in taking the correct decision. Thus, it is not always easy to detect the heart disease because it requires skilled knowledge or experiences about heart failure symptoms for an early prediction. Most of the medical dataset are dispersed, widespread and assorted. However, data mining is a robust technique for extracting invisible, predictive and actionable information from the extensive databases. In this paper, by using info gain feature selection technique and removing unnecessary features, different classification techniques such that KNN, Decision Tree (ID3), Gaussian Naïve Bayes, Logistic Regression and Random Forest are used on heart disease dataset for better prediction. Different performance measurement factors such as accuracy, ROC curve, precision, recall, sensitivity, specificity, and F1-score are considered to determine the performance of the classification techniques. Among them, Logistic Regression performed better, and the classification accuracy is 92.76% [18].

H. S. Niranjana Murthy et al. (2014) proposed a Dimensionality reduction using neuro-genetic approach for early prediction of coronary heart disease suggests that the development of a Neuro-genetic model for the prediction of coronary heart diseases. The novelty of this work is feature subset selection using multi-objective genetic algorithm without sacrificing the accuracy of ANN based heart disease predictor. Subsequently, the selected feature subset is used to predict the level of angiographic coronary heart disease using neural networks. The performance of the developed Neuro-Genetic model is evaluated using heart disease database obtained from Cleveland Clinic Foundation Database with all attributes are numeric-valued. The accuracy of the designed Neruo-Genetic model is validated using 303 patient data sets obtained for different age groups. This study exhibits early detection of heart disease with high testing accuracy of 89.58% through minimized feature subset, thereby reducing the complexity [7].

D. K. Ravish et al. (2014) proposed a Heart function monitoring, prediction and prevention of Heart Attacks: Using Artificial Neural Networks suggests that Heart Attacks are the major cause of death in the world today, particularly in India. The need to predict this is a major necessity for improving the country's healthcare sector. Accurate and precise prediction of the heart disease mainly depends on Electrocardiogram (ECG) data and clinical data. These data's must be fed to a non linear disease prediction model. This non linear heart function monitoring module must be able to detect arrhythmias such as tachycardia, bradycardia, myocardial infarction, atrial, ventricular fibrillation, atrial ventricular flutters and PVC's. In this they have developed an efficient method to acquire the clinical and ECG data, so as to train the Artificial Neural Network to accurately diagnose the heart and predict abnormalities if any. The overall process can be categorized into three steps. Firstly, they acquire the ECG of the patient by standard 3 lead pre jelled electrodes. The acquired ECG is then processed, amplified and filtered to remove any noise captured during the acquisition stage. This analog data is now converted into digital format by A/D converter, mainly because of its uncertainty. Secondly we acquire 4-5 relevant clinical data's like mean arterial pressure (MAP), fasting blood sugar (FBS), heart rate (HR), cholesterol (CH), and age/gender. Finally we use these two data's i.e. ECG and clinical data to train the neural network for classifying the heart disease and to predict abnormalities in the heart or it's functioning [5].

V Krishnaiah et al. (2014) proposed a Diagnosis of heart disease patients using fuzzy classification technique suggests that the various techniques in Data Mining have been applied to predict the heart disease patients. But, the uncertainty in data was not removed with the techniques available in data mining and implemented by various authors. To remove uncertainty of unstructured data, an attempt was made by introducing fuzziness in the measured data. A membership function was designed and incorporated with the measured value to remove uncertainty and fuzzified data was used to predict the heart disease patients. Further, an attempt was made to classify the patients based on the attributes collected from medical field. Minimum Euclidean distance Fuzzy K-NN classifier was designed to classify the training and testing data belonging to different classes.

It was found that Fuzzy K-NN classifier suits well as compared with other classifiers of parametric techniques [22].

2.2 EXISTING SYSTEM

Existing diagnose coronary heart disease (CHD) is based on your medical and family histories, your risk factors, a physical exam, and the results from tests and procedures. No single text can diagnose CHD. The doctor thinks you have CHD, they may recommend one or more of the following tests

2.2.1 EKG (Electrocardiogram)

The electrocardiogram, also referred to as ECG, 12-lead ECG, or EKG, is a non-invasive diagnostic test that evaluates your heart's electrical system to assess for heart disease. It uses flat metal electrodes placed on your chest to detect the electrical charges generated by your heart as it beats, which are then graphed. Your doctor can analyze the patterns to get a better understanding of your heart rate and heart rhythm, identify some types of structural heart disease, and evaluate cardiac efficiency.

An ECG detects your heart's electrical rhythm and produces what's known as a tracing, which looks like squiggly lines shown in Figure2.1. This tracing consists of representations of several waves that recur with each heartbeat, about 60 to 100 times per minute. The wave pattern should have a consistent shape. If your waves are not consistent, or if they do not appear as standard waves, this is indicative of heart disease.

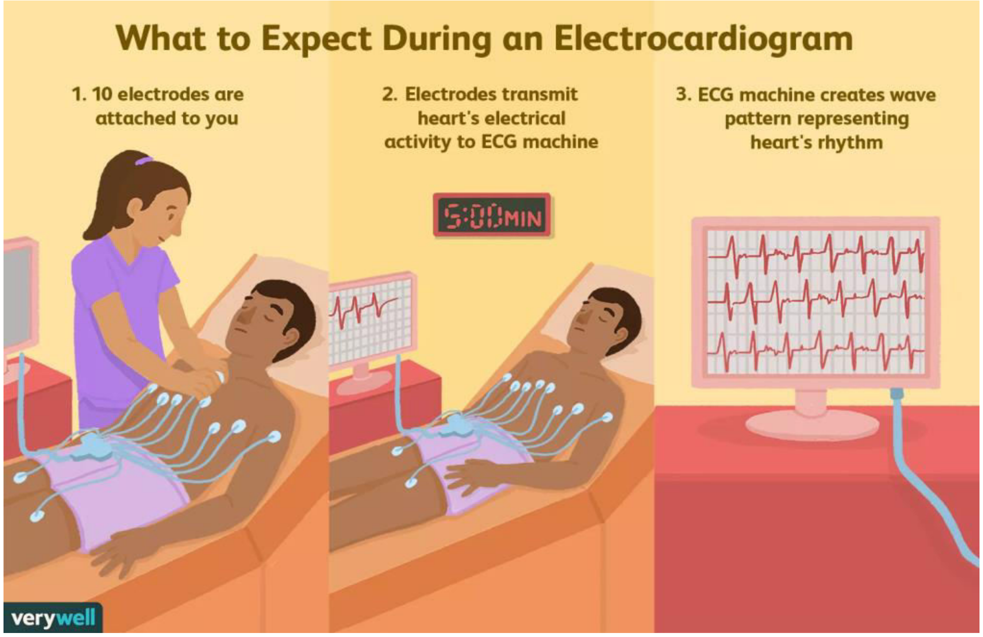

What to Expect During an Electrocardiogram
1. 10 electrodes are attached to you
2. Electrodes transmit heart's electrical activity to ECG machine
5:00MIN
3. ECG machine creates wave pattern representing heart's rhythm
verywell

Figure 2.1. Typical image of EKG

Limitations

The ECG is one of the most commonly used tests in medicine because it can screen for a large variety of cardiac conditions, the machines are readily available in most medical facilities, the test is simple to perform, is safe, and relatively inexpensive.

That said, an ECG has its limitations:

- The ECG reveals the heart rate and rhythm only during the few seconds it takes to record the tracing. If an arrhythmia (heart rhythm irregularity) occurs only intermittently, an ECG might not pick it up, and ambulatory monitoring may be required.
- The ECG is often normal or nearly normal with many types of heart disease, such as coronary artery disease.

- Sometimes, abnormalities that appear on the ECG turn out to have no medical significance after a thorough evaluation is done.

Stress Testing

During stress testing, you exercise to make your heart work hard and beat fast while heart tests are done. If you can't exercise, you may be given medicines to increase your heart rate.When your heart is working hard and beating fast, it needs more blood and oxygen. Plaque-narrowed coronary (heart) arteries can't supply enough oxygen-rich blood to meet heart's needs.

A stress test can show possible signs and symptoms of CHD, such as:

- Abnormal changes in heart rate or blood pressure
- Shortness of breath or chest pain
- Abnormal changes in heart rhythm

As part of some stress tests, pictures are taken of your heart while you exercise and while you rest. These imaging stress tests can show how well blood is flowing in your heart and how well your heart pumps blood when it beats.

2.2.2 Echocardiography

Echocardiography (echo) uses sound waves to create a moving picture of heart as shown in Figure 2.2. The test provides information about the size and shape of your heart and how well your heart chambers and valves are working.Echo also can show areas of poor blood flow to the heart, areas of heart muscle that aren't contracting normally, and previous injury to the heart muscle caused by poor blood flow.

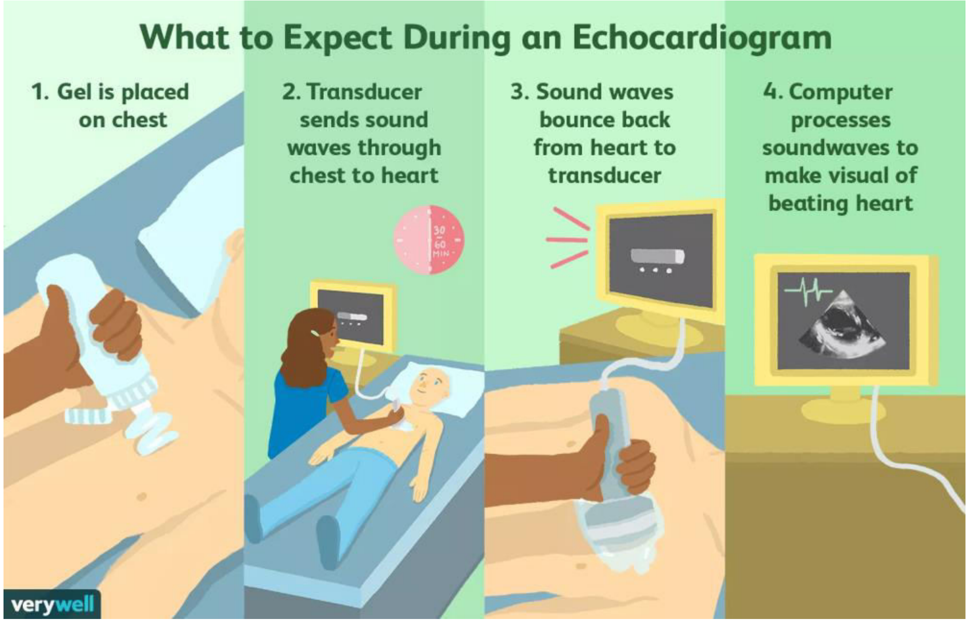

What to Expect During an Echocardiogram
1. Gel is placed on chest
2. Transducer sends sound waves through chest to heart
30 60 MIN
3. Sound waves bounce back from heart to transducer
4. Computer processes soundwaves to make visual of beating heart
verywell

Figure 2.2 Functionalities of ECG

Limitations

While the echocardiogram provides a lot of information about cardiac anatomy, it does not visualize the coronary arteries or blockages in your coronary arteries. If imaging the coronary arteries is necessary, a cardiac catheterization is commonly performed.

Certain physical variations, such as a thick chest wall or emphysema, can interfere with visualization of heart during an echocardiogram. If you have one of these conditions and need an echo, you might need an invasive ultrasound of heart known as a transesophageal echocardiogram (TEE).

CHAPTER 3
SYSTEM SPECIFICATION

3.1 Hardware Requirements

- Processor : Intel Dual-Core processor.
- RAM : 4 GB.
- HDD : 1 TB.

3.2 Software Requirements

- Operating System - Windows 7,8,8.1,xp.
- Documentation -MS Word, Google docs.
- Language – Java.

3.3 Software Tools

Tools used is WEKA 3.8 for classification. Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

Weka Knowledge Explorer

The Weka Knowledge Explorer is an easy to use graphical user interface that harnesses the power of the weka software. Each of the major weka packages Filters, Classifiers, Clusters, Associations, and Attribute Selection is represented in the Explorer along with a Visualization tool which allows datasets and the predictions of Classifiers and Clusters to be visualized in two dimensions.

It consists of the following panel:

- Pre-process Panel
- Classifier Panel
- Cluster Panel
- Associate Panel
- Select Attributes Panel
- Visualize Panel

CHAPTER 4

IMPLEMENTATION

4.1 Proposed system

Cardiovascular diseases are the most common cause of death worldwide over the last few decades in the developed as well as underdeveloped and developing countries. Early detection of cardiac diseases and continuous supervision of clinicians can reduce the mortality rate. In this Performance analysis of heart disease prediction using classification algorithms were implemented. For the accurate detection of the heart disease, an efficient machine learning technique should be used which had been derived from a distinctive analysis among several machine learning algorithms in a Java Based Open Access Data Mining Platform, WEKA. The proposed algorithm was validated using UCI machine learning dataset, where 10-fold cross-validation is applied in order to analyse the performance of heart disease detection. The following five algorithms are used for the classification SVM,MLP,KNN, Naïve Bayes and Random forest.

4.2 SYSTEM ARCHITECTURE

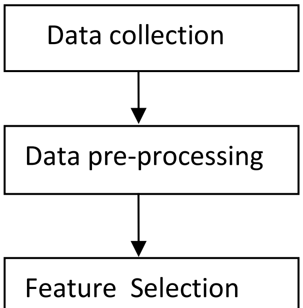


Figure 4.1 System architecture

Data cleaning: Data that need to be analyzed using machine learning algorithms can be incomplete, noisy and inconsistent. It also handles missing values of the attributes of interest where it replaces the appropriate mean value of the attribute. Similarly incorrect attribute values are cleared and manually filled with its mean value. Data are cleaned by manipulating the missing values, removing the outliers and smoothing the noisy data.

Data reduction: It is highly impractical to analyze the complex data and also it requires long time to process huge amount of data. Thus, Data reduction techniques are essential process in data classification, where it reduces the dimensionality of the data without compromising the integrity of original data. The reduction can be done either by reducing the quantity or the attributes. Several Feature selection techniques are available to remove the irrelevant and redundant attributes from the dataset. In our analysis, Information gain attribute evaluation based feature selection technique is used. The entropy of the attribute towards the class labels are analyzed and less significance attributes are removed for further classification. In UCI heart disease dataset, there exists 270 instances with 13 attributes, after applying information gain evaluation, only 7 attributes are considered for further classification.

4.3 LIST OF MODULES

The project would consists of the following modules :

1. Data collection
2. Data pre-processing
3. Feature Extraction
4. Train – Split
5. Classification

4.3.1 Data collection

Data collection largely consists of data acquisition, data labeling, and improvement of existing data or models. We have collected heart dataset from UCI Machine learning repository. The data consists of 270 instances and it does not have any missing values. It has 150 instance of absence of heart disease and 120 instance of presence of heart disease and also has 13 attributes which are real and nominal.

4.3.2 Data pre-processing

Real-world data is generally incomplete and noisy, and is likely to contain irrelevant and redundant information or errors . Data preprocessing, which is an important step in data mining processes, helps transform the raw data to an understandable format . Besides, some modeling techniques are quite sensitive to the predictors, such as linear regression. Thus, examining and preprocessing data before entering the model is essential. This chapter outlines some important methods in data preprocessing, including data cleaning, data transformation and data reduction.

4.3.2.1 Data cleaning

Dealing with missing data

Missing data is common in real world dataset, and it has a profound effect on the final analysis result which may make the conclusion unreliable. There are different types of missing data. We should have a good understanding of why the data is missing. If data is missing at random or if the missing is related to a particular predictor but the predictor has no relationship with the outcome, then the data sample can still represent the population. However, if the data is missing in a pattern that is related to the response, it can lead to a significant bias in the model, making the analysis result unreliable.

Many techniques have been proposed to deal with missing data, and generally they can be divided into two strategies. The first and the simplest one would be removing the missing data directly. If the missing data is distributed at random or the missing is related to a predictor that has zero correlation with the response, and the dataset is large enough, then the removal of missing data has little effect on the performance of analysis. However, if one of the above conditions is not satisfied, then simply removing the missing data is inappropriate.

The second strategy is to fill in or impute the missing data based on the rest of the data. Generally, there are two approaches. One method simply uses the average of the predictor to fill in the missing value. Alternatively, we can use a learning algorithm such as Bayes or decision tree to predict the missing value . It is worth noting that additional uncertainty is added by the imputation.

Dealing with outliers

Outlier is defined as an observation point that is distant from the mainstream data. The presence of outliers can break a model’s analysis ability. For example, outliers have a strong impact on data scaling and the regression fits. However, it is hard to identify outliers if the data range is not specified. One of the most efficient ways to identify outliers may be data visualization. By looking at a figure, we can point out some suspected observations and check whether these values are scientifically valid (e.g. positive weight). Removal of outliers can only be taken when there are truly valid reasons.

Instead of removing the outliers, an alternative way is to transform data to minimize the effect caused by outliers.

4.3.3 Feature Extraction

In the feature extraction technique, features or independent variables from the data set are transformed into new independent variables known as new feature space. Newly constructed feature space explains the data most and only significant data are selected. Let, there are n features of X1…….Xn. After feature extraction there are m attributes where (n > m) and this feature extraction is done with some mapping function, F.

InfoGainAttributeEval evaluates the worth of an attribute by measuring the information gain with respect to the class. InfoGain(Class,Attribute) = H(Class) – H(Class | Attribute), where H is the information entropy.

Attribute Name	Type
Chest pain type	Nominal
Maximum heart rate	Real

Exercise induced angina	Binary
Old peak = ST depression induced by exercise relative to rest	Real
Slope of the peak exercise	Ordered
Number of major vessels	Real
Thalassemia	Nominal

Table 4.1: List of selected attributes after feature selection

Table 1 consists of attributes which are selected after applying the feature selection algorithm i.e. Information gain attribute evaluation.

4.3.4 Train – Test split

To test our model we should split the data into train dataset and test dataset. We shall use the train dataset and then it will be tested on the test dataset. We train dataset in WEKA which automatically splits the train and test dataset.

For testing we can supply the test data separately by inputting it in a arff format.

4.3.5 Classification

Classification can be performed on structured or unstructured data. Classification is a technique where we categorize data into a given number of classes. The main goal of a classification problem is to identify the category/class to which a new data will fall under.

Machine Learning Algorithms

For the purpose of comparative analysis, five Machine Learning algorithms are discussed. The different Machine Learning (ML) algorithms are k-Nearest Neighbour (KNN), Random Forest (RF), Support Vector Machine (SVM), Naïve Bayes and Multilayer perceptron(MLP). The reason to choose these algorithms is based on their popularity

1 k-Nearest Neighbour

Nearest Neighbour algorithms are among the simplest of all machine learning algorithms. The idea is to memorize the training set and then to predict the label of any new instance on the basis of the labels of its closest neighbours in the training set. The rationale behind such a method is based on the assumption that the features that are used to describe the domain points are relevant to their labelling in a way that makes close-by points likely to have the same label. Furthermore, in some situations, even when the training set is immense, finding a nearest neighbour can be done extremely fast.

Mathematically, p: X∗X → ℝ where Ψ is a function that returns the distance between the two points of X(xi, xi'). The Euclidean distance between two points can be calculated by following formula:

$$p\left(x, x^{'}\right) = \left|x - x^{'}\right| = \sqrt{\sum_{i=1}^{d}\left(xi - xi^{'}\right)^{2}}$$

k in the k-nearest neighbour is the number of the data points closest to the new instance. For example, if k =1 then the algorithm will choose the nearest one instance or if k = 4, then the algorithm will choose the closest four neighbour instances and will classify them accordingly. The idea can be better illustrated with figure 4.2

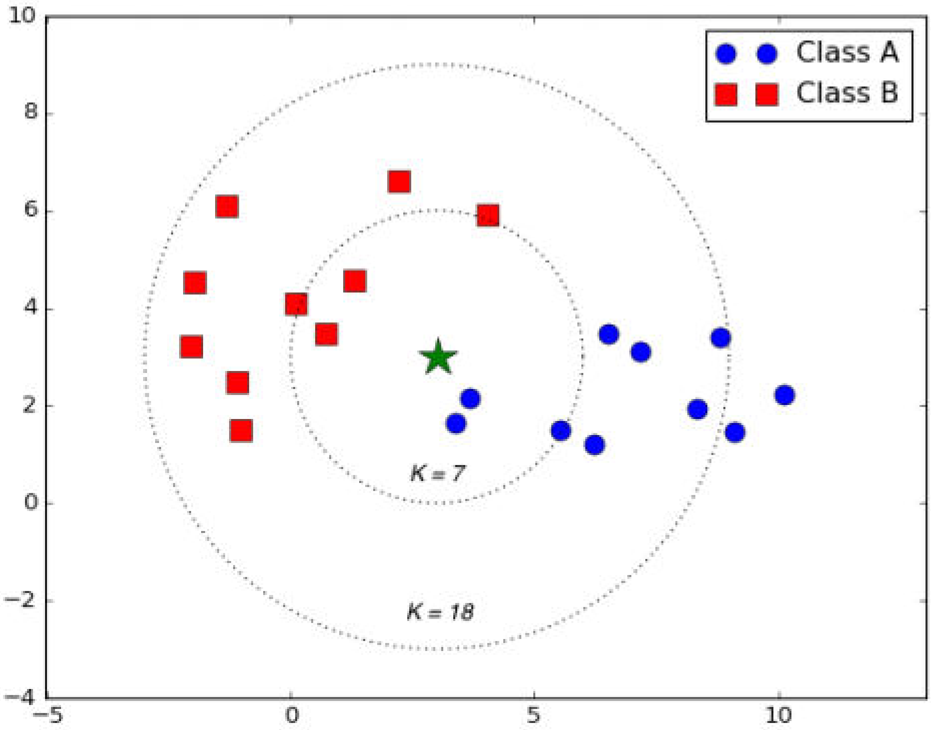


Figure 4.2 k- Nearest Neighbour algorithm

Referring to Figure 4.1, Green star is the data point to be classified, blue circles are class A data points and red squares are class B rectangles. The Euclidean distance between the green star and all other red and blue points are measured. The star will be classified to the data points which have least distance. If k =7, then distance between all the seven points are measured from the star and star will be classified to the data points with least distance, in this case with blue data point.

1) Find the 'k' instances in the data set that are

closest to 'S'

2) These 'k' instances then vote to determine

the class of 'S'

The Algorithm

- Let ‘K’ be the no. of nearest neighbors and ‘T’ be the set of training examples.
    - For each test sample $z=(x_1,y_1)$ do
    - Compute $d(x_1, x)$, the distance between ‘Z’ and every sample
    - Select T_z subset symbol, the set of k closest training examples to z

      $y1=argmax\sum (xi,yi)$

2 Support Vector Machine

(SVM) A Support Vector Machine (SVM) is a classifier which distinct the various classes of data by the use of a hyper-plane. SVM is modelled with the training data and it outputs the hyper-plane in the test data. The SVM model tries to find the space in the matrix of data where different classes of data can be widely differentiated and draws a hyper-plane.

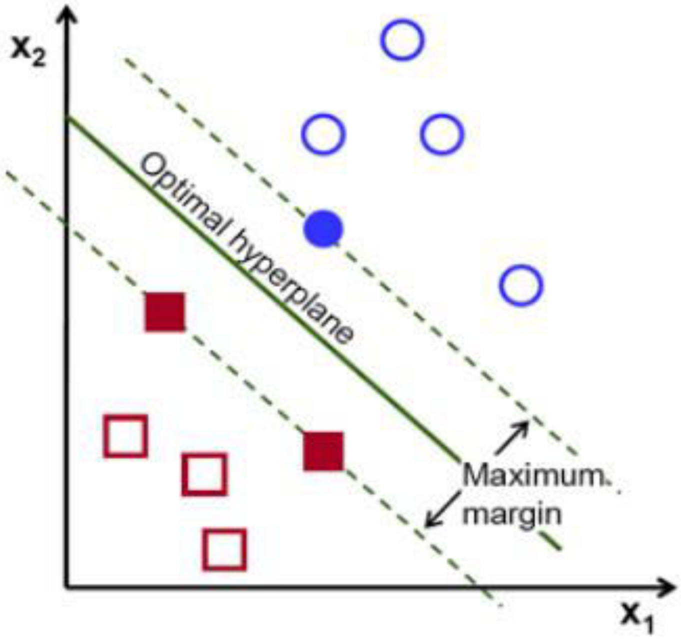


Figure 4.3 Support Vector Machine

In Figure 4.3, Red and Blue are the classes of labelled training data points. To classify them linearly a hyper-plane can be drawn but the question is: There is more than one way to draw a hyper-plane so which one is optimal? An optimal hyper-plane is chosen which maximizes the margin between the classes. Hyper-plane need not always belinear. A hyper-plane in SVM can also work as a non-linear classifier using technique known as kernel-trick.

Pseudo code

```
● Initialize : α i = 0; f i = -y i
● Compute b high , I high , b low and I low
● Update α Ihigh and α Ilow
● repeat
● Update f i
● until b low &lt;= b up +2τ
● Update the threshold ‘b’
● Store new α 1 and α 2 values
● Update weight vector ‘w’, if SVM is linear
```

1 Multilayer perceptron

A multilayer perceptron (MLP) is a class of feedforward artificial neural network. A MLP consists of, at least, three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training.[1][2] Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

Algorithm

- Assign initial weights to attributes
- Add up all weighted inputs and check with sigmoid threshold to compute output
- Check for error in output / any hidden layer If yes

  Change the weights

  Repeat step 2 and 3 Else

  Exit
- Algorithm recursively continues until all the data is classified

Layers

The MLP consists of three or more layers (an input and an output layer with one or more hidden layers) of nonlinearly-activating nodes. Since MLPs are fully connected, each node in one layer connects with a certain weight *wij* to every node in the following layer.

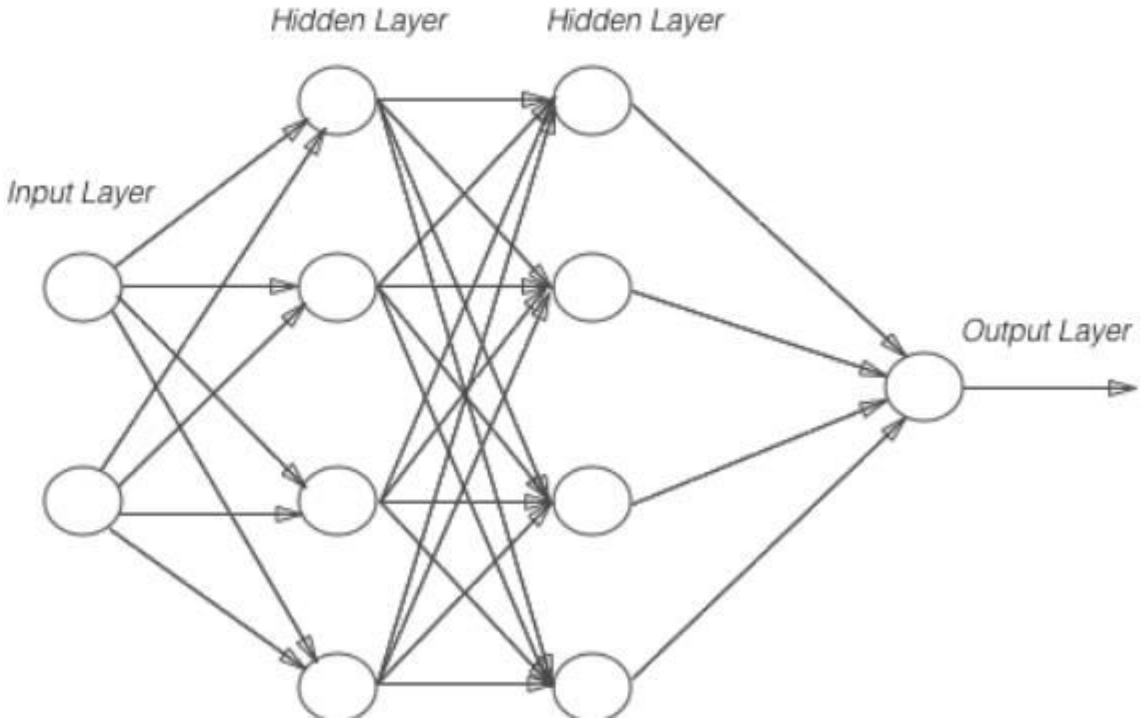


Figure 4.4 Schematic of Neural Network

4 Naïve Bayes

Naïve Bayes or Naïve Bayes classifier in a machine learning context is a classifier which uses the Bayes theorem to classify the data and it assumes that the probability of certain feature X is totally independent of another feature Y [7,397]. Bayes theorem can be easily explained with the following example. Probability of spanners produced by machine A is 0.6, machine B is 0.4. A defect in spanners in the whole of production is 1 percent and the probability of defected spanners produced by machine A is 50 percent and machine B is 50 percent. In this scenario Bayes theorem can be used to answer what is the probability of a defected machine produced by machine B is ?

$$P(Defect \mid Machine\ B) = \frac{P(Machine\ B \mid Defect) * P(Defect)}{P(Machine\ B)}$$

Bayes theorem provides a way to calculate the probability of the hypothesis given that there is prior knowledge about the problem.

$$Posterior = \frac{Likelihood * Prior}{Evidence}$$

There are three types of Naïve Bayes. Gaussian Naïve Bayes, Multinomial Naïve Bayes and Bernoulli Naïve Bayes. Gaussian Naïve Bayes is used in classification problems, Multinomial Naïve Bayes is used in multinomial distributed data and Bernoulli Naïve Bayes is used in data with multivariate Bernoulli distribution.

5 Random Forest

A random forest is a classifier consisting of a collection of decision trees, where each tree is constructed by applying an algorithm A on the training set S and an additional random vector, θ where θ is sampled i.i.d.(independently and identically distributed) from some distribution. The prediction of the random forest is obtained by a majority vote over the predictions of the individual trees .

The Random Forest algorithm works in following steps:

1. Picks random K data points from the training data.
2. Builds a decision tree for these K data points.
3. Chooses the Ntree subset from the trees and performs step 1 and step 2
4. Decides the category or result on the basis of the majority of votes.

To understand Random Forest more intuitively it is better to understand decision trees and they can be better understood with the help of a diagram

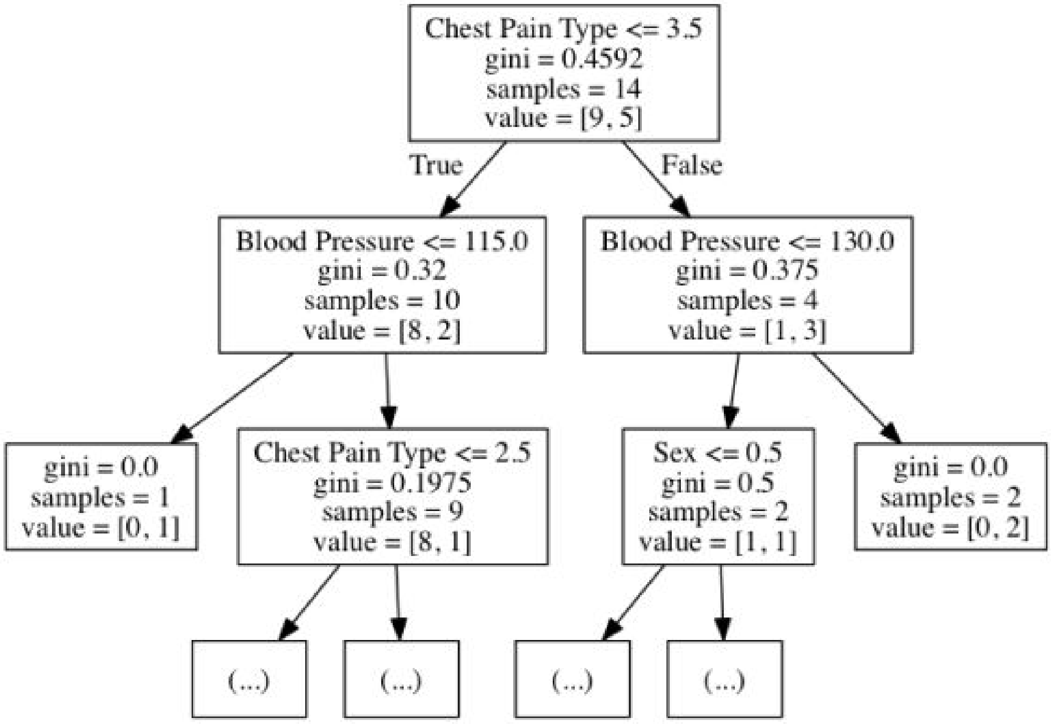


Figure 4.5 Schematic of Random Forest

Referring to Figure 4.5, it illustrates how decision trees work. If decision trees are used to predict whether there is a heart disease or not, then it will decide according to the above mapping. Gini is the coefficient of the spread of the data, samples are the number of data taken for classifying the node, value is the array of samples which are classified as true or false.

Algorithm

- Construct the probability function(sigmoid)
- From the training set, select a new bootstrap sample.
- Grow on unpruned tree on this bootstrap sample.
- Randomly select (*m try*) at each internal node and determine best split.
- If each tree is fully grown. Do not perform pruning.
- Output overall prediction as the majority vote from all the trees.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Introduction

Machine Learning is a field which is raised out of Artificial Intelligence(AI). Applying AI, we wanted to build better and intelligent machines. For the purpose of comparative analysis, five Machine Learning algorithms are discussed. The different Machine Learning (ML) algorithms are k-Nearest Neighbour (KNN), Random Forest (RF), Support Vector Machine (SVM), Naïve Bayes and Multilayer perceptron(MLP). The reason to choose these algorithms is based on their popularity

5.1.1 Performance Metrics

The performance analysis of heart disease classification by using different algorithms is listed below:

Accuracy

Accuracy or predictive accuracy is the measure of the proportion of instances that are correctly classified. It shows how close the predicted value is to the true value or the theoretical value. Formula for calculating the accuracy is given below.

$$\text{Accuracy} = \frac{TP + TN}{TP+FP+TN+FN} \tag{1}$$

Accuracy is the measure of how close or near the predicted value is to the actual or theoretical value. For example, if the actual value of a person’s height is 6 feet and measured value or predicted value is 5.9 then it is quite an accurate measurement.

Precision

Precision is defined as the proportion of true positive instances which are classified as positive. It shows how close predicted values are to each other. Formula for calculating precision is given below.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2}$$

Precision is the measure of how close or near the predicted values are to each other. For example, if actual value of person’s height is 6 feet and measured value is or predicted value is 5.5 and every time measurement is taken height is 5.4 or 5.6 then prediction is quite precise but not accurate.

Recall

Recall is the measure number of positive predictions divided by number of positive class values in the test data. Recall is the completeness of the classifiers and it can be calculated with below equation

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3}$$

Recall is simply the measure of how many true samples are predicted from the all the samples.

F-Measure

F-measure conveys the balance between the precision and recall. The balanced F-score is the harmonic mean of both precision and recall and it can be calculated as

$$\text{F-Measure} = \frac{2*\text{Recall}*\text{Precision}}{\text{Precision+Recall}} \quad (4)$$

MCC

MCC is a correlation coefficient used as measure of quality of binary. It generally varies between -1 and +1. -1 indicates the total number of disagreement between actuals and prediction, 1 indicates a total number of agreements between actuals and predictions. 0 indicates no better than random prediction.

Receiver operating curve (ROC)

ROC is graphical representation of true positive rate against false positive rate The different classifier algorithms are imposed on the pre-processed data.

Confusion matrix helps us to evaluate total number of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) instance. With the help of TP, TN, FP and FN value, it is possible us to validate the various performance measures such as accuracy, precision , recall, F-measure, ROC, MCC, etc

5.1.2 Performance Evaluation

Algorithm	**Accuracy %**	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F-Measur**	**MCC**	**ROC**	**PRC**

						e			
SVM	85.92	0.859	0.153	0.860	0.859	0.858	0.715	0.853	0.807
KNN	78.88	0.789	0.222	0.788	0.789	0.788	0.571	0.782	0.726
Naïve Bayes	85.18	0.852	0.160	0.853	0.852	0.851	0.699	0.887	0.881
MLP	81.48	0.815	0.201	0.816	0.815	0.813	0.624	0.846	0.841
Random Forest	79.62	0.796	0.213-	0.796	0.796	0.796	0.586	0.871	0.876

Table 5.1: Study of five algorithms

Table 5.1 shows the comparative study of five different algorithms with different performance metrics. Here accuracy of SVM is better than that of all the five algorithms.

Momentum-1.0	**Momentum-100**	**Momentum-0.2**	**Momentum-0.3**
70.37%	70.37	81.48	82.59
0.74	0.74	0.815	0.826
0.347	0.347	0.2	0.818
0.727	0.727	0.816	0.827
0.74	0.74	0.815	0.826
0.685	0.685	0.814	0.825
0.408	0.408	0.624	0.646

0.676	0.676	0.848	0.849
0.635	0.635	0.85	0.844

Table 5.2: MLP with constant learning rate and varying momentum

Table 5.2 shows the performance metrics of Multilayer perceptron with constant learning rate and varying momentum. The value of momentum lies between 0 and 1. Momentum helps in smoothing out the variations, if the gradient keeps changing direction. A right value of momentum can be either learned by hit and trial or through cross-validation.

Algorithm	**Learning Rate= 0.3**	**Learning Rate=1**	**Learning Rate=0.2**
Accuracy	81.48%	82.59%	80.00%
TP Rate	0.815	0.826	0.8
FP Rate	0.201	0.191	0.212
Precision	0.816	0.828	0.8
Recall	0.815	0.826	0.8
F-measure	0.813	0.824	0.799
MCC	0.624	0.647	0.593
ROC	0.846	0.837	0.839
PRC	0.841	0.827	0.835

Table 5.3: MLP with constant momentum and different learning rate

Table 5.3 shows the performance metrics of MLP with different learning rate. While training of Perceptron minima needs to be determined. If we choose larger value of learning rate then we might overshoot that minima and smaller values of learning rate might take long time for convergence.

Algorithm	C=1.0	C=2.0	C=0.1
Accuracy	85.92%	84.44%	83.33%
TP Rate	0.859	0.844	0.833
FP Rate	0.153	0.166	0.175
Precision	0.86	0.845	0.833
Recall	0.859	0.844	0.833
F-measure	0.858	0.844	0.833
MCC	0.715	0.684	0.662
ROC	0.853	0.839	0.829
PRC	0.807	0.789	0.777

Table 5.4: SMO with constant kernel function and varying C

Table 5.4 shows the performance metrics of SMO.C is a regularization parameter that controls the trade off between the achieving a low training error and a low testing error. The C parameter tells the SVM optimization how much you want to avoid misclassifying each training example. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

Algorithm	puk	RBF kernal	Normalized poly kernal
Accuracy	82.22%	84.07%	84.07%

Tp Rate	0.822	0.841	0.841
FP Rate	0.191	0.181	0.172
Precision	0.823	0.847	0.842
Recall	0.822	0.841	0.841
F-measure	0.821	0.838	0.84
MCC	0.639	0.681	0.677
ROC	0.816	0.83	0.834
PRC	0.763	0.783	0.784

Table 5.5: SMO with constant C and different Kernel

Table 5.5 shows the performance metrics of SMO with constant C and different kernel function.

Algorithm	**K=1**	**K=3**	**K=5**	**K=7**
Accuracy	78.88%	80.74%	81.48%	81.10%
TP Rate	0.789	0.87	0.815	0.811
FP Rate	0.222	0.199	0.196	0.21
Precision	0.788	0.807	0.815	0.811

Recall	0.789	0.807	0.815	0.811
F-measure	0.788	0.807	0.814	0.81
MCC	0.571	0.609	0.624	0.616
ROC	0.782	0.852	0.869	0.877
PRC	0.726	0.811	0.838	0.851

Table 5.6: KNN with Euclidean distance and varying K

Table 5.6 shows performance metrics of KNN with Euclidean distance and varying K. If k is even number, the accuracy is less than the condition of odd. When the k value increases the accuracy also increases and also when k is too large the might be a time delay. Euclidean distance is the most widely used distance function.

Algorithm	**K=1**	**K=3**	**K=5**	**K=7**
Accuracy	78.14%	80.00%	80.74%	81.85%
TP Rate	0.781	0.8	0.87	0.819
FP Rate	0.23	0.28	0.27	0.197
Precision	0.781	0.8	0.88	0.82
Recall	0.781	0.8	0.87	0.819
F-measure	0.781	0.8	0.806	0.817

MCC	0.556	0.594	0.69	0.632
ROC	0.774	0.842	0.862	0.632

Table 5.7: KNN with Manhattan distance and varying K

Table 5.7 shows the performance metrics of KNN with Manhattan distance and different K. When the k value increases the accuracy also increases. Euclidean distance or Manhattan distance is used to calculate the accuracy that are almost the lowest and under a different ratio of training and testing data

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this paper, we carried out an experiment to find the predictive performance of different classifiers. We select five popular classifiers considering their qualitative performance for the experiment. We also choose one dataset from heart available at UCI machine learning repository. Support Vector Machine is the best in performance. In order to compare the classification performance of four machine learning algorithms, Classifiers are applied on same data and results are compared on the basis of misclassification and correct classification rate and according to experimental results in table 1, it can be concluded that Support Vector Machine is the best as compared to Multilayer perceptron ,K-Nearest Neighbour, Random forest and Naïve Bayes.

6.2 Future Work

After analysing the quantitative data generated from the computer simulations, Moreover their performance is closely competitive showing slight difference. So, more experiments on several other datasets need to be considered to draw a more general conclusion on the comparative performance of the classifiers.

CHAPTER 7

REFERENCES

[1] Aakash Chauhan et al. “Heart Disease Prediction using Evolutionary Rule Learning” , 2018 4th International Conference on Computational Intelligence & Communication Technology (CICT)”, 04 October 2018.

[2] Aditi Gavhane et al. “Prediction of Heart Disease Using Machine Learning” 2nd International conference on Electronics, Communication and Aerospace

Technology, DOI: 10.1109/ICECA.2018.8474922,2018.

[3] A H Chen et al. “HDPS: Heart disease prediction system” , 2011 Computing in Cardiology, 09 March 2012.

[4] Ankita Dewan et al. “Prediction of Heart Disease Using a Hybrid Technique in Data Mining Classification”, 2nd InternationalConference on Computingfor SustainableGlobal Development (INDIACom), 04 May 2015.

[5] D. K. Ravish et al. “Heart function monitoring, prediction and prevention of Heart Attacks: Using Artificial Neural Networks”, 2014 International Conference on Contemporary Computing and Informatics (IC3I),DOI: 10.1109/IC3I.2014.7019580, 26 January 2015.

[6] H. Mai et al. “Non-Laboratory-Based Risk Factors for Automated Heart Disease Detection”, 2018 12th International Symposium on Medical Information and Communication Technology (ISMICT),DOI: 10.1109/ISMICT.2018.8573706, 13 December 2018.

[7] H. S. Niranjana Murthy et al. “Dimensionality reduction using neuro-genetic approach for early prediction of coronary heart disease”, International Conference on Circuits, Communication, Control and Computing, DOI: 10.1109/CIMCA.2014.7057817, 12 March 2015.

[8] Jayshril S. Sonawane et al. “Prediction of heart disease using learning vector quantization algorithm”, 2014 Conference on IT in Business, Industry and Government (CSIBIG), DOI: 10.1109/CSIBIG.2014.7056973, 12 March 2015.

[9] Lalaantika Sharma et al. “Classification and development of tool for heart diseases (MRI images) using machine learning”, 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), DOI: 10.1109/PDGC.2016.7913149, 27 April 2017

[10] M.A.Jabbar et al. “Heart disease prediction system based on hidden naïve bayes classifier”, 2016 International Conference on Circuits, Controls, Communications and Computing (I4C), DOI: 10.1109/CIMCA.2016.8053261, 02 October 2017.

[11] Meenal Saini_ et al. “Prediction of heart disease severity with hybrid data mining”, 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), DOI: 10.1109/TEL-NET.2017.8343565, 23 April 2018.

[12] N. K. Salma Banu et al. “Prediction of heart disease at early stage using data mining and big data analytics: A survey”, 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT),DOI: 10.1109/ICEECCOT.2016.7955226

[13]N. Prabakaran et al. “Prediction of Cardiac Disease Based on Patient's Symptoms” , 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), DOI: 10.1109/ICICCT.2018.8473271, 27 September 2018.

[14] Rashmi G Saboji et al. “A scalable solution for heart disease prediction using classification mining technique” “ 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)” DOI:

10.1109/ICECDS.2017.8389755, 21 June 2018

[15] Rifki Wijaya et al. “ Preliminary design of estimation heart disease by using machine learning ANN within one year”, 2013 Joint International Conference on Rural Information & Communication Technology and Electric-Vehicle Technology (rICT & ICeV-T) ,08 May 2014.

[16]R.W.Jones, et al. “Prediction of heart disease using neural network”, International Conference on Computer Science and Engineering (UBMK) 2017.

[17] Sana Shaikh et al. “Electronic recording system-heart disease prediction system”, 2015 International Conference on Technologies for Sustainable Development (ICTSD), DOI: 10.1109/ICTSD.2015.7095854,30 April 2015.

[18] S. M. M. Hasan et al. “Comparative Analysis of Classification Approaches for Heart Disease Prediction”, 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2),DOI: 10.1109/IC4ME2.2018.8465594,20 September 2018.

[19] Tahira Mahboob . “Evaluating ensemble prediction of coronary heart disease using receiver operating characteristics”, 2017 Internet Technologies and Applications (ITA), DOI: 10.1109/ITECHA.2017.8101920, 09 November 2017.

[20] Tülay KarayÕlan et al.“Prediction of Heart Disease Using Neural Network”, 2017 International Conference on Computer Science and Engineering (UBMK),DOI: 10.1109/UBMK.2017.8093512, 02 November 2017.

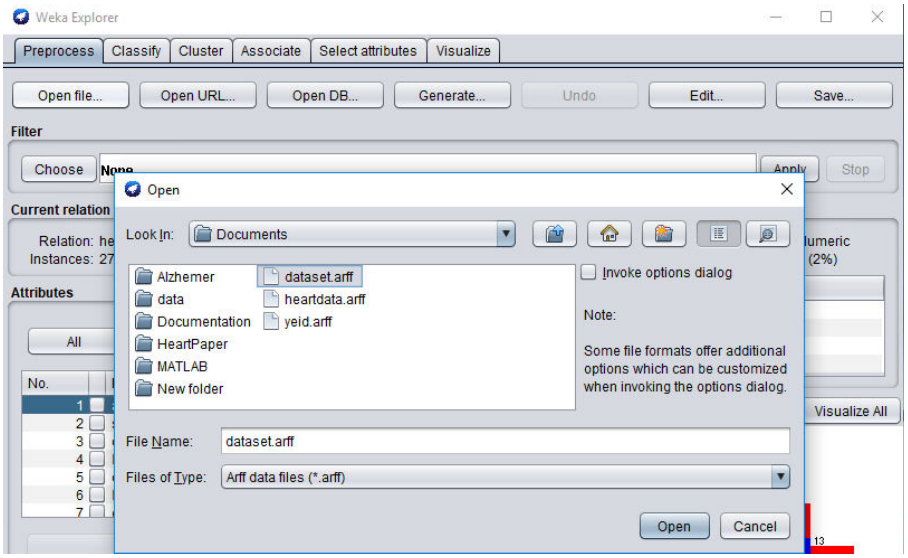
[21] UCI Machine Learning Repository [online]. URL:https://archive.ics.uci.edu/ml/machine-learning-databases/heartdisease/heart-disease.names Accessed 26 February 2017.

[22] V Krishnaiah et al. “Diagnosis of heart disease patients using fuzzy classification technique” ,International Conference on Computing and Communication Technologies,DOI: 10.1109/ICCCT2.2014.7066746,26 March 2015.

CHAPTER 8

APPENDIX

A.1 Screen shots



Look In: Documents

Note:

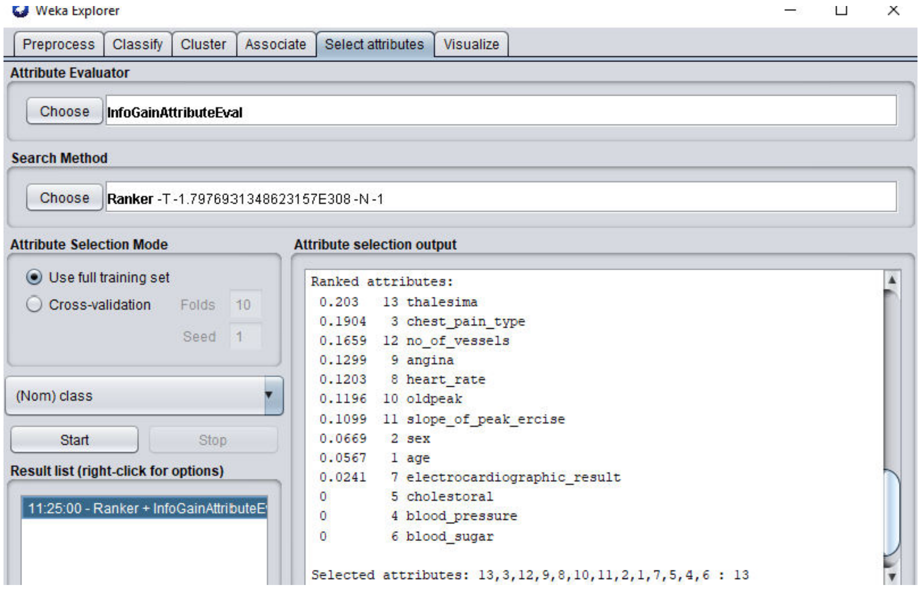
Some file formats offer additional options which can be customized when invoking the options dialog.

File Name: dataset.arff

Files of Type: Arff data files (*.arff)

Figure A.1.1 Importing the data

Figure A.1.1 shows that opening the file to import the dataset into the WEKA and it should be in the format of arff.



Attribute Evaluator: InfoGainAttributeEval

Search Method: Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10 Seed 1

(Nom) class

Result list (right-click for options)

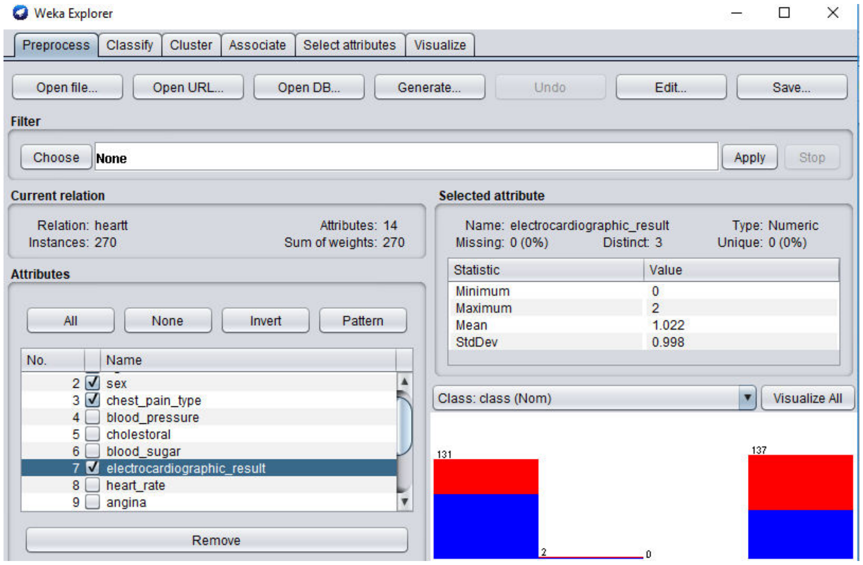
11:25:00 - Ranker + InfoGainAttributeE

Attribute selection output

```
Ranked attributes:
 0.203    13 thalesima
 0.1904    3 chest_pain_type
 0.1659   12 no_of_vessels
 0.1299    9 angina
 0.1203    8 heart_rate
 0.1196   10 oldpeak
 0.1099   11 slope_of_peak_ercise
 0.0669    2 sex
 0.0567    1 age
 0.0241    7 electrocardiographic_result
 0         5 cholestoral
 0         4 blood_pressure
 0         6 blood_sugar

Selected attributes: 13,3,12,9,8,10,11,2,1,7,5,4,6 : 13
```

Figure A.1.2 Feature Selection

Figure A.1.2 shows that the feature selection by using the InfoGainAttributeEval (Information Gain Attribute Evaluation) and Ranker search for selecting the best feature which is ranked by using ranker search method.



Filter: None

Current relation

Relation: heartt
Instances: 270
Attributes: 14
Sum of weights: 270

Selected attribute

Name: electrocardiographic_result
Type: Numeric
Missing: 0 (0%)
Distinct: 3
Unique: 0 (0%)

Statistic	Value
Minimum	0
Maximum	2
Mean	1.022
StdDev	0.998

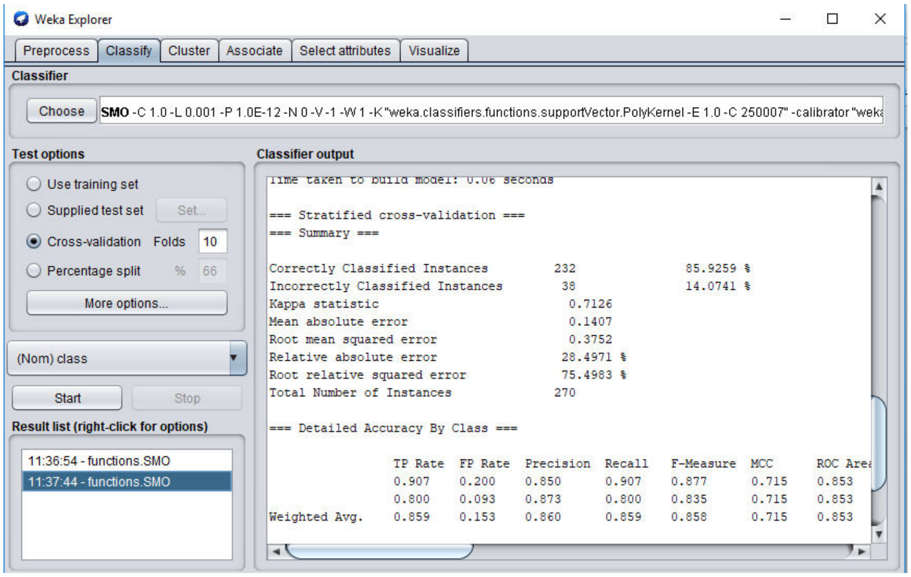
Class: class (Nom)

Attributes

No.	Name
2	☑ sex
3	☑ chest_pain_type
4	□ blood_pressure
5	□ cholestoral
6	□ blood_sugar
7	☑ electrocardiographic_result
8	□ heart_rate
9	□ angina

Figure A.1.3 Removing lowest ranked attributes

Figure A.1.3 shows that the removal of lowest ranked attributes by selecting and removing them.



Classifier

SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka

Test options

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

Result list (right-click for options)

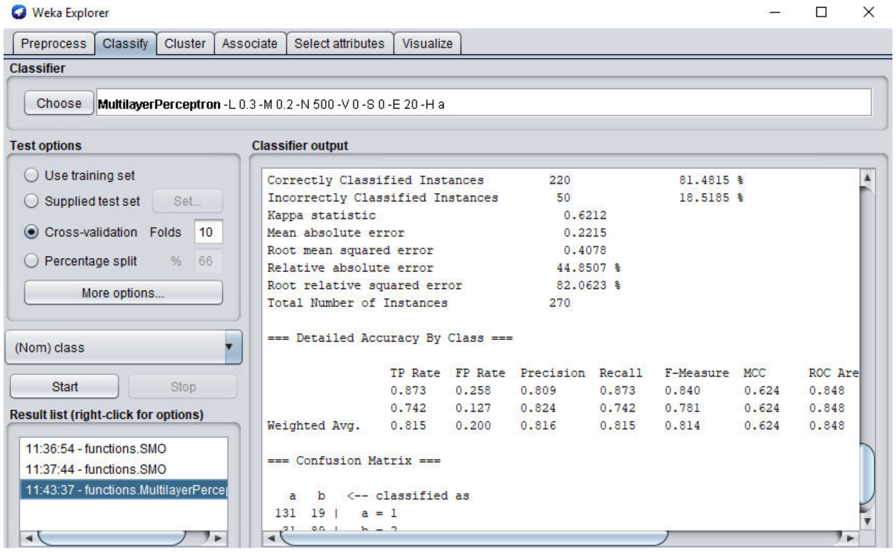
11:36:54 - functions.SMO
11:37:44 - functions.SMO

Classifier output

```
Time taken to build model: 0.06 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         232               85.9259 %
Incorrectly Classified Instances        38               14.0741 %
Kappa statistic                          0.7126
Mean absolute error                      0.1407
Root mean squared error                  0.3752
Relative absolute error                 28.4971 %
Root relative squared error             75.4983 %
Total Number of Instances              270

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area
                 0.907    0.200    0.850      0.907    0.877      0.715    0.853
                 0.800    0.093    0.873      0.800    0.835      0.715    0.853
Weighted Avg.    0.859    0.153    0.860      0.859    0.858      0.715    0.853
```

Figure A.1.4 Classification using SMO

Figure A.1.4 shows that the classification of dataset by using SMO. Accuracy of correctly classified instance is 85.92% % and incorrectly classified instance is 18.51% which is best among all other algorithms.



MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

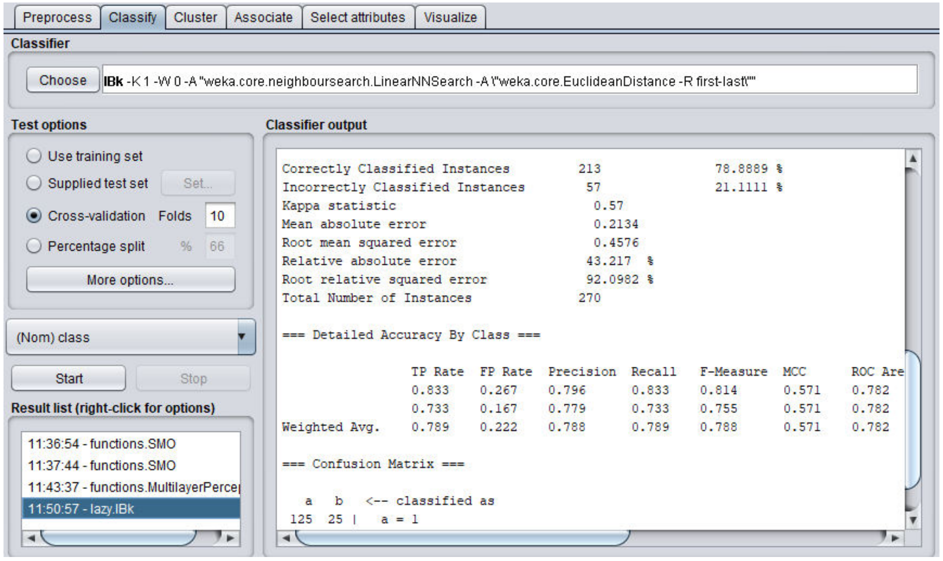
(Nom) class

11:36:54 - functions.SMO
11:37:44 - functions.SMO
11:43:37 - functions.MultilayerPerce

```
Correctly Classified Instances         220               81.4815 %
Incorrectly Classified Instances        50               18.5185 %
Kappa statistic                          0.6212
Mean absolute error                      0.2215
Root mean squared error                  0.4078
Relative absolute error                 44.8507 %
Root relative squared error             82.0623 %
Total Number of Instances              270

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Are
                 0.873    0.258    0.809      0.873    0.840      0.624    0.848
                 0.742    0.127    0.824      0.742    0.781      0.624    0.848
Weighted Avg.    0.815    0.200    0.816      0.815    0.814      0.624    0.848

=== Confusion Matrix ===

   a   b   <-- classified as
 131  19 |   a = 1
  31  89 |   b = 2
```

Figure A.1.5 Classification using MLP

Figure A.1.5 shows that the classification of dataset by using MLP. Accuracy of correctly classified instance is 81.48% and incorrectly classified instance is 18.51%.



IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

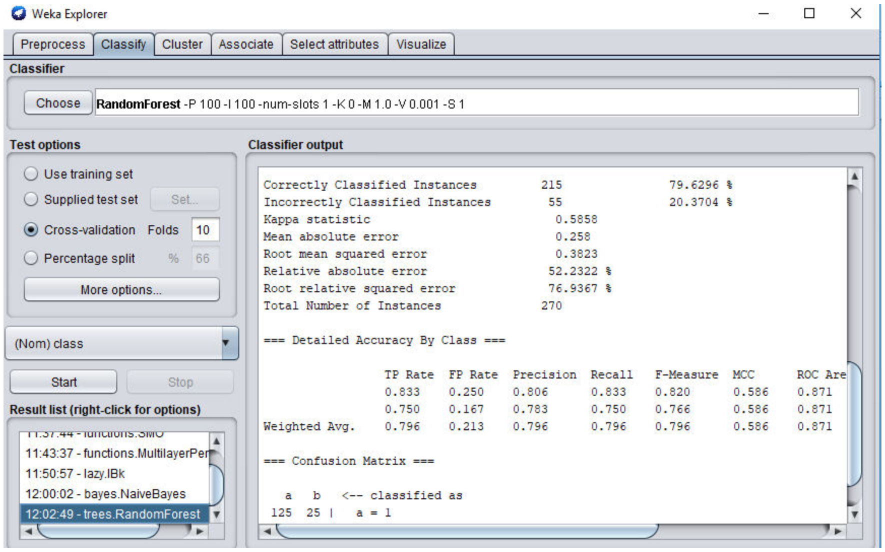
(Nom) class

11:36:54 - functions.SMO
11:37:44 - functions.SMO
11:43:37 - functions.MultilayerPerce
11:50:57 - lazy.IBk

```
Correctly Classified Instances         213               78.8889 %
Incorrectly Classified Instances        57               21.1111 %
Kappa statistic                          0.57
Mean absolute error                      0.2134
Root mean squared error                  0.4576
Relative absolute error                 43.217  %
Root relative squared error             92.0982 %
Total Number of Instances              270

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Are
                 0.833    0.267    0.796      0.833    0.814      0.571    0.782
                 0.733    0.167    0.779      0.733    0.755      0.571    0.782
Weighted Avg.    0.789    0.222    0.788      0.789    0.788      0.571    0.782

=== Confusion Matrix ===

   a   b   <-- classified as
 125  25 |   a = 1
```

Figure A.1.6 Classification using KNN

Figure A.1.6 shows that the classification of dataset by using KNN. Accuracy of correctly classified instance is 78.88% and incorrectly classified instance is 21.11%.



RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

```
11:37:44 - functions.SMO
11:43:37 - functions.MultilayerPer
11:50:57 - lazy.IBk
12:00:02 - bayes.NaiveBayes
12:02:49 - trees.RandomForest
```

```
Correctly Classified Instances         215               79.6296 %
Incorrectly Classified Instances        55               20.3704 %
Kappa statistic                          0.5858
Mean absolute error                      0.258
Root mean squared error                  0.3823
Relative absolute error                 52.2322 %
Root relative squared error             76.9367 %
Total Number of Instances              270

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Are
                 0.833    0.250    0.806      0.833    0.820      0.586    0.871
                 0.750    0.167    0.783      0.750    0.766      0.586    0.871
Weighted Avg.    0.796    0.213    0.796      0.796    0.796      0.586    0.871

=== Confusion Matrix ===

   a   b   <-- classified as
 125  25 |   a = 1
```

Figure A.1.7 Classification using Random Forest

Figure A.1.8 shows that the classification of dataset by using Naïve Bayes. Accuracy of correctly classified instance is 79.62% and incorrectly classified instance is 20.37%.

A.2 Coding

```
package weka.classifiers.functions;


import weka.classifiers.AbstractClassifier;
import weka.classifiers.Classifier;
import weka.classifiers.functions.supportVector.Kernel;
import weka.classifiers.functions.supportVector.PolyKernel;
import weka.classifiers.functions.supportVector.SMOset;
import weka.core.Attribute;
import weka.core.Capabilities;
import weka.core.Capabilities.Capability;
import weka.core.DenseInstance;
import weka.core.Instance;
import weka.core.Instances;
import weka.core.Option;
import weka.core.OptionHandler;
import weka.core.RevisionUtils;
import weka.core.SelectedTag;
import weka.core.Tag;
import weka.core.Utils;
import weka.filters.unsupervised.attribute.NominalToBinary;
import weka.filters.unsupervised.attribute.Normalize;
import weka.filters.unsupervised.attribute.ReplaceMissingValues;
import weka.filters.unsupervised.attribute.Standardize;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Enumeration;
import java.util.Random;
import java.util.Vector;

public void buildClassifier(Instances insts) throws Exception {

  if (!m_checksTurnedOff) {
   // can classifier handle the data?
   getCapabilities().testWithFail(insts);

   // remove instances with missing class
   insts = new Instances(insts);
   insts.deleteWithMissingClass();

   /* Removes all the instances with weight equal to 0.
    MUST be done since condition (8) of Keerthi's paper
    is made with the assertion Ci > 0 (See equation (3a). */
   Instances data = new Instances(insts, insts.numInstances());
   for (int i = 0; i < insts.numInstances(); i++) {
     if (insts.instance(i).weight() > 0)
       data.add(insts.instance(i));
   }
   if (data.numInstances() == 0) {
     throw new Exception("No training instances left after removing " +
             "instances with weight 0!");
   }
   insts = data;
 }

  if (!m_checksTurnedOff) {
   m_Missing = new ReplaceMissingValues();
   m_Missing.setInputFormat(insts);
   insts = Filter.useFilter(insts, m_Missing);
  } else {
   m_Missing = null;
  }

  if (getCapabilities().handles(Capability.NUMERIC_ATTRIBUTES)) {
   boolean onlyNumeric = true;
```

```
 if (!m_checksTurnedOff) {
  for (int i = 0; i < insts.numAttributes(); i++) {
   if (i != insts.classIndex()) {
    if (!insts.attribute(i).isNumeric()) {
     onlyNumeric = false;
     break;
    }
   }
  }
 }

 if (!onlyNumeric) {
  m_NominalToBinary = new NominalToBinary();
  m_NominalToBinary.setInputFormat(insts);
  insts = Filter.useFilter(insts, m_NominalToBinary);
 } else {
  m_NominalToBinary = null;
 }
} else {
 m_NominalToBinary = null;
}

if (m_filterType == FILTER_STANDARDIZE) {
 m_Filter = new Standardize();
 m_Filter.setInputFormat(insts);
 insts = Filter.useFilter(insts, m_Filter);
} else if (m_filterType == FILTER_NORMALIZE) {
      m_Filter = new Normalize();
      m_Filter.setInputFormat(insts);
      insts = Filter.useFilter(insts, m_Filter);
    } else {
      m_Filter = null;
    }

    m_classIndex = insts.classIndex();
    m_classAttribute = insts.classAttribute();
    m_KernelIsLinear = (m_kernel instanceof PolyKernel) && (((PolyKernel)
m_kernel).getExponent() == 1.0);

    // Generate subsets representing each class
    Instances[] subsets = new Instances[insts.numClasses()];
    for (int i = 0; i < insts.numClasses(); i++) {
      subsets[i] = new Instances(insts, insts.numInstances());
    }
    for (int j = 0; j < insts.numInstances(); j++) {
      Instance inst = insts.instance(j);
      subsets[(int) inst.classValue()].add(inst);
    }
    for (int i = 0; i < insts.numClasses(); i++) {
      subsets[i].compactify();
    }
  /**
   * Estimates class probabilities for given instance.
   *
```

```
 * @param inst the instance to compute the probabilities for
 * @throws Exception in case of an error
 */
public double[] distributionForInstance(Instance inst) throws Exception {

  // Filter instance
  if (!m_checksTurnedOff) {
    m_Missing.input(inst);
    m_Missing.batchFinished();
    inst = m_Missing.output();
  }

  if (m_NominalToBinary != null) {
    m_NominalToBinary.input(inst);
    m_NominalToBinary.batchFinished();
    inst = m_NominalToBinary.output();
  }

  if (m_Filter != null) {
    m_Filter.input(inst);
    m_Filter.batchFinished();
    inst = m_Filter.output();
  }

  if (!m_fitCalibratorModels) {
    double[] result = new double[inst.numClasses()];
    for (int i = 0; i < inst.numClasses(); i++) {
```

```
    for (int j = i + 1; j < inst.numClasses(); j++) {
      if ((m_classifiers[i][j].m_alpha != null) ||
              (m_classifiers[i][j].m_sparseWeights != null)) {
        double output = m_classifiers[i][j].SVMOutput(-1, inst);
        if (output > 0) {
          result[j] += 1;
        } else {
          result[i] += 1;
        }
      }
    }
  }
  Utils.normalize(result);
  return result;
} else {

  // We only need to do pairwise coupling if there are more
  // then two classes.
  if (inst.numClasses() == 2) {
    double[] newInst = new double[2];
    newInst[0] = m_classifiers[0][1].SVMOutput(-1, inst);
    newInst[1] = Utils.missingValue();
    DenseInstance d = new DenseInstance(1, newInst);
    d.setDataset(m_classifiers[0][1].m_calibrationDataHeader);
    return m_classifiers[0][1].m_calibrator.distributionForInstance(d);
  }
  double[][] r = new double[inst.numClasses()][inst.numClasses()];
  double[][] n = new double[inst.numClasses()][inst.numClasses()];
  for (int i = 0; i < inst.numClasses(); i++) {
   for (int j = i + 1; j < inst.numClasses(); j++) {
    if ((m_classifiers[i][j].m_alpha != null) ||
        (m_classifiers[i][j].m_sparseWeights != null)) {
     double[] newInst = new double[2];
     newInst[0] = m_classifiers[i][j].SVMOutput(-1, inst);
     newInst[1] = Utils.missingValue();
     DenseInstance d = new DenseInstance(1, newInst);
     d.setDataset(m_classifiers[i][j].m_calibrationDataHeader);
     r[i][j] = m_classifiers[i][j].m_calibrator.distributionForInstance(d)[0];
     n[i][j] = m_classifiers[i][j].m_sumOfWeights;
    }
   }
  }
  return weka.classifiers.meta.MultiClassClassifier.pairwiseCoupling(n, r);
 }
}
public int[] obtainVotes(Instance inst) throws Exception {

 // Filter instance
 if (!m_checksTurnedOff) {
  m_Missing.input(inst);
  m_Missing.batchFinished();
  inst = m_Missing.output();
 }
```

```
  if (m_NominalToBinary != null) {
    m_NominalToBinary.input(inst);
    m_NominalToBinary.batchFinished();
    inst = m_NominalToBinary.output();
  }

  if (m_Filter != null) {
    m_Filter.input(inst);
    m_Filter.batchFinished();
    inst = m_Filter.output();
  }

  int[] votes = new int[inst.numClasses()];
  for (int i = 0; i < inst.numClasses(); i++) {
    for (int j = i + 1; j < inst.numClasses(); j++) {
      double output = m_classifiers[i][j].SVMOutput(-1, inst);
      if (output > 0) {
        votes[j] += 1;
      } else {
        votes[i] += 1;
      }
    }
  }
  return votes;
}

public String[] getOptions() {
```

```
Vector<String> result = new Vector<String>();

if (getChecksTurnedOff())
  result.add("-no-checks");

result.add("-C");
result.add("" + getC());

result.add("-L");
result.add("" + getToleranceParameter());

result.add("-P");
result.add("" + getEpsilon());

result.add("-N");
result.add("" + m_filterType);

if (getBuildCalibrationModels())
  result.add("-M");

result.add("-V");
result.add("" + getNumFolds());

result.add("-W");
result.add("" + getRandomSeed());
```

```
  result.add("-K");
  result.add("" + getKernel().getClass().getName() + " " +
Utils.joinOptions(getKernel().getOptions()));

  result.add("-calibrator");
  result.add(getCalibrator().getClass().getName() + " "
      + Utils.joinOptions(((OptionHandler)getCalibrator()).getOptions()));

  Collections.addAll(result, super.getOptions());

  return (String[]) result.toArray(new String[result.size()]);
 }

 /**
  * Disables or enables the checks (which could be time-consuming). Use with
  * caution!
  *
  * @param value if true turns off all checks
  */
 public void setChecksTurnedOff(boolean value) {
  if (value)
   turnChecksOff();
  else
   turnChecksOn();
 }
  public String kernelTipText() {
  return "The kernel to use.";
```

```
}

/**
 * sets the kernel to use
 *
 * @param value the kernel to use
 */
public void setKernel(Kernel value) {
  m_kernel = value;
}

/**
 * Returns the kernel to use
 *
 * @return                    the current kernel
 */
public Kernel getKernel() {
  return m_kernel;
}
```